Three Methods for Occupation Coding Based on Statistical Learning

Gweon, Hyukjun; Schonlau, Matthias; Kaczmirek, Lars; Blohm, Michael; Steiner, Stefan

DE GRUYTER
OPEN

Journal of Official Statistics, Vol. 33, No. 1, 2017, pp. 101–122, http://dx.doi.org/10.1515/JOS-2017-0006

Three Methods for Occupation Coding Based on Statistical Learning

Hyukjun Gweon[1], *Matthias Schonlau*[1], *Lars Kaczmirek*[2], *Michael Blohm*[2], *and Stefan Steiner*[1]

Occupation coding, an important task in official statistics, refers to coding a respondent's text answer into one of many hundreds of occupation codes. To date, occupation coding is still at least partially conducted manually, at great expense. We propose three methods for automatic coding: combining separate models for the detailed occupation codes and for aggregate occupation codes, a hybrid method that combines a duplicate-based approach with a statistical learning algorithm, and a modified nearest neighbor approach. Using data from the German General Social Survey (ALLBUS), we show that the proposed methods improve on both the coding accuracy of the underlying statistical learning algorithm and the coding accuracy of duplicates where duplicates exist. Further, we find defining duplicates based on ngram variables (a concept from text mining) is preferable to one based on exact string matches.

Key words: Automated coding; Machine learning; ISCO-88; ALLBUS.

1. Introduction

Classifying a respondent's occupation is essential in official statistics and social science research. It enables the international comparison of the official statistics on occupation and work and is the starting point for numerous status scales or prestige measures. It is a "foundation of much, if not most research on social stratification" (Ganzeboom and Treiman 2003, 159) and social inequality. Because occupation is a risk factor in many diseases, classifying occupations is an important first step for epidemiological analyses, industrial hygiene, and other biomedical sciences.

There are quite a few different classification schemes, but all have hundreds of occupation codes and the codes are always nested in hierarchies. For example, the International Standard Classification of Occupations 1988 (ISCO-88) (Elias 1997) is a classification of four nested levels characterized by four digits. The first digit distinguishes nine major groups, and an undifferentiated tenth major group for the Armed Forces. There are 28 sub-major groups (two-digit combinations), 116 minor groups (three-digit

[1] Department of Statistics and Actuarial Science, University of Waterloo, 200 University Avenue West, Waterloo, Ontario N2L 3G1 Canada. Emails: hgweon@uwaterloo.ca, schonlau@uwaterloo.ca, and shsteiner@uwaterloo.ca.
[2] GESIS – Leibniz-Institute for the Social Sciences, PO Box 12 21 55, D-68072 Mannheim, Germany. Emails: lars.kaczmirek@gesis.org and michael.blohm@gesis.org
Acknowledgments: This research was supported in part by the Social Sciences and Humanities Research Council of Canada (SSHRC # 435-2013-0128) (PI: Schonlau) and we are grateful for the support.

combinations) and 390 unit groups (four-digit combinations). Table 1 gives coding for sub-major group 71, extraction and building trades workers.

To ascertain a survey respondent's occupation, typically an open-ended question is asked (Belloni et al. 2014). Alternative ways to find a respondent's occupation include the use of search trees in web surveys (Tijdens 2014, 2015), but open-end questions are most common. The main example in this article is the biannual ALLBUS survey (ALLBUS 2015) conducted by GESIS – Leibniz Institute for the Social Sciences. The ALLBUS survey uses open-ended questions to ask about occupation (Scholz and Wasmer 2009). Using multiple choice questions to elicit four-digit occupation codes is not sensible because there are too many codes, and more importantly, respondents often would not know how to classify themselves because occupation coding rules are complex (International Labour Office 1990; Geis 2011; Elias 1997; Belloni et al. 2014).

Traditionally, assigning an occupation code to each answer text has been conducted manually by human coders. Manual coding is time-consuming and expensive, requiring professional knowledge. Occupation coding is also difficult: there are hundreds of predefined occupation codes and even more occupation titles. For example, the ISCO-88 classification contains 390 four-digit occupation codes. Another difficulty is that coding even by professional coders may be inconsistent. The coding quality of a record depends on the length of the occupation description as well as the difficulty of the words in the record (Conrad et al. 2016).

Table 1. ISCO-88 Sub-Major Group 71: extraction and building trades workers.

71	Extraction and building trades workers	
	711	Miners, shotfirers, stone cutters and carvers
		7111 Miners and quarry workers
		7112 Shotfirers and blasters
		7113 Stone splitters, cutters and carvers
	712	Building frame and related trades workers
		7121 Builders
		7122 Bricklayers and stonemasons
		7123 Concrete placers, concrete finishers and related workers
		7124 Carpenters and joiners
		7129 Building frame and related trades workers not elsewhere classified
	713	Building finishers and related trades workers
		7131 Roofers
		7132 Floor layers and tile setters
		7133 Plasterers
		7134 Insulation workers
		7135 Glaziers
		7136 Plumbers and pipe fitters
		7137 Building and related electricians
		7139 Building finishers and related trade workers not elsewhere classified
	714	Painters, building structure cleaners and related trades workers
		7141 Painters and related workers
		7143 Building structure cleaners

In an attempt to partially automate coding, researchers have implemented various rule-based coding schemes. For example, if the text answer contained a word matching an entry in a predefined dictionary, then the corresponding code in the dictionary was assigned. More recently, statistical learning or machine learning approaches have been employed: a model is trained on manually coded training data and is then used to predict the most probable code for new data (Statistical learning and machine learning are synonymous for the purpose of this article. For brevity we just use the phrase "statistical learning" for the remainder of the article). This approach is favored, for example, by the Australian Bureau of Statistics (Clarke and Brooker 2011). Autocoders based on statistical learning have also been developed in the United States (Day 2014) and in Germany (Bethmann et al. 2014).

Although the automated methods reduce costs for occupation coding, fully automated coding remains challenging. With partial automatic coding, easy-to-code answers are coded automatically, and-hard-to-code answers are coded manually. A measure of confidence – a numerical score – is used to distinguish between easy-to-code and hard-to-code text answers (Scholtus et al. 2014). For example, the CASCOT system proposes manual coding when a score for the coding quality drops below a modifiable threshold (Jones and Elias 2004).

In this article we consider three new techniques for improving automated coding:

(a) a combination of two statistical learning models for different levels of aggregation,
(b) a combination of a duplicate-based approach with a statistical learning one, and
(c) a modified nearest neighbor approach.

The remainder of this article is organized as follows: In Section 2 we give background on approaches to automated occupation coding. In Section 3, we introduce the three techniques for improving automated coding. In Section 4, we evaluate the proposed approaches with data from the 2006 German ALLBUS survey coded by GESIS based on ISCO-88 codes. In Section 5, we conclude with a discussion.

2. Automated Occupation Coding

This section gives an overview of how to evaluate the performance in automated occupation coding, as well as two types of commonly used approaches: rule-based approaches and approaches based on statistical learning. The new approaches we introduce in this article are mostly based on statistical learning.

2.1. Production Rate and Accuracy

When some answer texts are coded automatically and some are coded manually, a score or a probability is needed to distinguish between hard-to-code and easy-to-code answers. All new records with scores above a threshold are coded automatically; all others are coded manually. The threshold is set according to the desired combination of accuracy and production rate. The production rate is the proportion of observations that can be coded automatically. For a given production rate, accuracy is the proportion of codes that are coded correctly. Note that there is a tradeoff between accuracy and production rate. High accuracy can be achieved for a small number of easy-to-code records. However, as the

production rate increases and more difficult answers are included, accuracy tends to decrease. The tradeoff relationship was illustrated in Chen et al. (1993).

2.2. *Preprocessing*

Before automated coding begins, text is often preprocessed. There is no standardized way of preprocessing, but there are a range of options, such as lower or upper casing all letters, removing duplicate blank spaces, automatically correcting spelling errors, removing very common words (so-called stopwords), and, less common in occupation coding but common in text mining, reducing words to their grammatical root (stemming). Preprocessing is an attempt to reduce the noise in the data.

2.3. *Rule-Based Occupation Coding*

If the text answer meets a prespecified logical condition (e.g., presence of a certain word) a specific code is assigned. Such "if-then" statements are called rules. Rules are written by experts or can be based on previous data analysis. Rules can be combined using boolean logic. Any one rule-based coding scheme consists of hundreds of rules leading to large dictionaries or look-up tables. Schierholz (2014) reports that this approach rarely codes more than 50% of records accurately. A variation on rule-based methods is to assign a score in favor of a category. If a text answer matches a rule, evidence can accumulate for multiple codes. In the end, the text answer is classified into the occupation code with the highest score. One of the earliest references to rule-based coding is O'Reagan (1972).

Rule-based systems are implemented in many institutions: the Washington State Department of Health (Ossiander and Milham 2006), the 1970 U.S. Population and Housing Census (Knaus 1987), the 1991 census data for Croatia and Bosnia-Herzegovina (Kalpic 1994), and the AIOCS system at the U.S. Census Bureau (Appel and Hellerman 1983; Chen et al. 1993). Statistics Canada further developed the AIOCS system and created the G-Code (formerly ACTR) software (Wenzowski 1988; Tourigny and Moloney 1995), which was also used for Italian census data (Ferrillo et al. 2008). The University of Warwick has a popular tool for automatic categorization called CASCOT (Jones and Elias 2004; see also Elias and Birch 2010 for performance of CASCOT), which has also been adapted to the Dutch language (Belloni et al. 2014).

2.4. *Occupation Coding Based on Statistical Learning*

Statistical models learn from already classified training data. Such methods can be used not only for occupation coding but also for general classification problems. Once the model has been trained, other observations can be classified automatically.

To build a model, text is first converted to numerical data. The standard text mining approach is to create a variable for each word that occurs in any of the answer texts. These unigram variables or one-grams either record the frequency of the word occurring in an answer text or simply the presence or absence of the word from the given answer text (Weiss et al. 2010; Joachims 1998). There are many different variations of this text mining approach, adding variables for the presence or absence of multi-word sequences (ngram variables), removing highly used words (stopwords) because they are probably not useful,

and stemming words to their grammatical root. The large number of variables are modeled with black-box statistical learning algorithms, such as support vector machines (*SVM*) (Vapnik 2000). The model may incorporate additional variables if available.

Different learning algorithms have been used for occupation coding. The Australian Bureau of Statistics (ABS) employed fully automatic categorization using support vector machines to code data from the 2006 Australian Census (Clarke and Brooker 2011). The ABS uses the Australian and New Zealand Standard Classification of Occupation (ANZSCO) scheme. To our knowledge this system is still in use by the ABS.

The American Community Survey (ACS) uses a variation on text mining (Thompson et al. 2012). Variables created from the text include one-word and two-word sequences (called "wordbits") as well as the full text. To limit the number of variables for analysis, a rareness threshold of 30 is used (i.e., the text has to occur at least 30 times before it is used as a variable). To further limit the number of variables for analysis, the corresponding text has to be "associated with a single industry/occupation code at least 50% of the time". The remaining variables, as well as variables like age and gender, are fed into a logistic regression. The code with the highest probability obtained by the logistic regression is assigned to a new record.

Some authors have investigated a nearest neighbor strategy, which assigns the code of the answer in the training data most closely resembling the answer in question. Different similarity metrics have been employed to measure nearness or resemblance between two answers. The PACE system employed the k nearest neighbor method with weighted feature metrics and reported accuracy 0.86 at production rate 0.57 for the U.S. Census Bureau data (Creecy et al. 1992). Jung et al. (2008) used cosine similarity but found this did not work well, possibly because they were working in Korean, a language quite different from languages with roots in Latin. Russ et al. (2014) used the nearest neighbor approach with a Jaccard similarity measure for classifying text answers into the Standard Occupational Classification (SOC) scheme. Coding by the nearest neighbour approach was considered correct if it agreed with one or both of the codes provided by the two human coders. The accuracy, that is, the proportion of correctly classified observations, for fully automated coding was 0.51 at the six-digit level and 0.64 at the three-digit level.

The ALWA survey at the German Institute for Employment Research (IAB) used the five-digit German national classification KldB 2010 (Schierholz 2014). The approach presented in Schierholz (2014) used the full preprocessed verbatim answer text rather than the text mining approach using ngram variables. Preprocessing included converting special German characters into regular ones, stripping leading and trailing spaces. Using verbatim answers (rather than ngrams) drastically reduced the number of variables for learning. Schierholz (2014) then experimented with various methods including Naive Bayes and a gradient boosting model (Friedman 2001). The experiment concluded that boosting and the Bayesian approaches performed similarly when high accuracy was desired.

3. Three Methods for Automated Occupation Coding

We first explain the duplicate method, a simple automated coding approach based on duplicate training observations. Next, we propose three new methods for automated

occupation coding. The first of these methods, combining statistical learning models at different levels of aggregation, is later also incorporated with the second method, resulting in two versions of the second method. For statistical learning models, any method that outputs probabilities can be used. In Section 4, we choose Support Vector Machines (Vapnik 2000) for our application.

For each method, the predicted occupation code is the code that has the highest score.

3.1. The Duplicate Method With the Ngram-Based Definition of Duplicates

An exact-string duplicate refers to two strings that are identical. Simple string preprocessing could improve performance and leads to what we call a preprocessed-string duplicate. Preprocessing the string might consist, for example, of lower-casing allletters and removing leading and trailing blanks. For example "Apotheker" (pharmacist), "apotheker" and " apotheker" would be considered duplicates after preprocessing.

We introduce a different definition of duplicates based on ngram variables: an ngram duplicate refers to a training observation with a text answer that has the same ngram representation (i.e., the same values for the variables created from the text). This is slightly different than an observation with the identical text answer. For example, the answer "Verwaltungsangestellte im Krankenhaus" (administrator in the hospital) and "Verwaltungsangestellte in einem Krankenhaus" (administrator in a hospital) are not identical texts. However, since "in", "im" and "einem" are stopwords and stopwords are removed, these two strings contain the same unigrams ("Verwaltungsangestellte", "Krankenhaus").

Suppose that there exist some duplicates of a new input record $\mathbf{x}$. Let $m_i(\mathbf{x})$ be the number of training duplicates having code c_i ($i = 1,2,\ldots,L$). We estimate the probability $p_d(c_i|\mathbf{x})$ based on the relative frequency of the training duplicates having code c_i:

$$\hat{p}_d(c_i|\mathbf{x}) = \begin{cases} \dfrac{m_i(\mathrm{x})}{M(\mathrm{x})} & \text{if } \mathrm{M(x)} > 0 \\ \dfrac{1}{L} & \text{otherwise} \end{cases},$$

where $M(\mathrm{x}) = \sum_{i=1}^{L} m_i(\mathbf{x})$ is the number of duplicates of $\mathbf{x}$ found in the training date. If no duplicate is found, the method assigns equal probability to each class. The code with the highest probability is chosen as the predicted code. The duplicate method leads to high accuracy for duplicates, although not to 100% accuracy, since coders try to resolve ambiguous situations with additional undocumented information or due to human error.

3.2. Combining Models from Different Levels of Aggregation

As seen in Table 1, occupation codes have a hierarchical structure. The ISCO-88 occupation codes consist of four-digit numbers. For example, the code 7131 (roofers) is part of the minor group 713 (Building finishers and related trades workers). Three-digit group codes aggregate related occupations. We propose to apply statistical learning separately to the four-digit unit occupation codes and to the three-digit group codes, and to combine probabilities as explained in the next paragraph. The motivation is as follows: Given the large number of occupation codes, the number of observations at the four-digit

level can be sparse. The number of observations will be relatively less sparse at the three-digit level. If classification from a four-digit classifier results in a near tie of occupation codes with different minor groups (different third digit), the evidence from the three-digit classifier may sway the classification to the correct four-digit code.

Suppose that code c_i $(i = 1, \ldots, L)$ belongs to a three-digit minor group m_j $(j = 1, \ldots, l)$ where L and l are the numbers of the four-digit and three-digit group codes respectively. Denote the probabilities from the statistical learning model for three-digits and four-digits as $\hat{p}_{3digit}(m_j|\mathbf{x})$ and $\hat{p}_{4digit}(c_i|\mathbf{x})$ for a record $\mathbf{x}$, respectively. We average the two probabilities:

$$\hat{p}_{3/4digit}(c_i|\mathbf{x}) = \frac{\hat{p}_{3digit}(m_j|\mathbf{x}) + \hat{p}_{4digit}(c_i|\mathbf{x})}{2}. \tag{1}$$

This averaging approach will also break ties at the four-digit level, unless the tied codes have the same three-digit code. A recent review of hierarchical classification methods in general (Silla and Freitas 2011), does not contain the proposed method. However, the proposed method may be viewed as a member of the local-classifier-per-level approaches as it fits a classifier for each three-digit and four-digit level independently.

3.3. A Hybrid Approach: Combining Duplicate and Statistical Learning Approaches

The proposed hybrid approach combines the approach based on duplicates in the training data with a statistical learning approach.

Let $\hat{p}_s(c_i|x)$ be the estimated probability obtained by a statistical learning approach. For the hybrid approach we define a combined score $\theta(c_i|\mathbf{x})$ as

$$\theta(c_i|\mathbf{x}) = \frac{M(\mathbf{x})}{M(\mathbf{x}) + 1} \cdot \hat{p}_d(c_i|\mathbf{x}) + \frac{1}{M(\mathbf{x}) + 1} \cdot \hat{p}_s(c_i|\mathbf{x}) \tag{2}$$

If there are no duplicates, the score equals the probability from the statistical learning approach $\hat{p}_s(c_i|\mathbf{x})$. When there are duplicates, coding by the duplicate method is desirable, as it leads to high accuracy. Hence, in the hybrid approach the statistical learning algorithm only influences the prediction when there is a tie among different duplicate codes. Equation (2) assigns the statistical learner a weight equivalent to that of a single duplicate, and the single duplicate is downweighted by the probability $\hat{p}_s(c_i|\mathbf{x}) < 1$.

When the production rate is less than 100%, the easier-to-learn new records are categorized automatically. The statistical learning algorithms also influence this prioritization of new records. When two new records each have the same number of duplicates and if $\hat{p}_d(c_i|\mathbf{x})$ is the same in each case, the record with the larger $\hat{p}_s(c_i|\mathbf{x})$ is assigned a greater $\theta(c_i|\mathbf{x})$ and therefore is prioritized for lower production rates.

We call this approach "hybrid-4digit" when $p_s(c_i|\mathbf{x})$ in Equation (2) is estimated using the statistical learning model for four-digit occupation codes, $\hat{p}_{4digit}(c_i|\mathbf{x})$. Subsection 3.2 defined $\hat{p}_{3/4digit}(c_i|x)$ in Equation (1), which combined two statistical learning models from different levels of aggregation. This idea can also be applied here. We call this approach "hybrid-3/4digit" when $p_s(c_i|\mathbf{x})$ in Equation (2) is estimated using $\hat{p}_{3/4digit}(c_i|x)$.

3.4. A Modified Nearest Neighbor Approach

The nearest neighbour approach (*NN*) (Fix and Hodges 1951) is another method employed in the occupation coding. *NN* classification finds a new record's nearest neighbor in the training data and also assigns the occupation code of that nearest neighbor to the new record. There can be multiple nearest neighbors (Yu 2002). *NN* can be viewed as a generalization of the duplicate approach: duplicates are nearest neighbors with a distance of zero. To define "near", a measure of distance, or, equivalently, a measure of similarity is needed. For text classification, cosine similarity is widely used (Knaus 1987; Iezzi et al. 2014; Maitra and Ramler 2010). Cosine similarity between two vectors $\mathbf{u}$ and $\mathbf{v}$ is defined as

$$\text{cosine}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}||\mathbf{v}|} = \frac{\sum u_i v_i}{\sqrt{\sum u_i^2}\sqrt{\sum v_i^2}}. \tag{3}$$

where $\mathbf{u}$ and $\mathbf{v}$ are vector representations of presence or absence of ngrams in the text. Similarity ranges from 0 to 1 depending on the degree of the similarity between two records. Similarity is 0 if two records have no common words and 1 if the two records are identical (in the sense of having the same ngram representation). When duplicates exist, the *NN* method predicts the code of records with similarity 1, which is equivalent to the duplicate method.

As before, we may want to only code easy-to-code text answers and leave difficult ones for manual coding. Hence, we propose to use a score that assigns a higher value to *NN* predictions that are believed to be more accurate. Given a new text input $\mathbf{x}$, denote $K(\mathbf{x})$ the number of nearest neighbors in the training data and $s(\mathbf{x})$ the similarity of the nearest neighbors. (Often $K(\mathbf{x}) > 1$ when multiple observations are the nearest neighbors.) Suppose that $k_i(\mathbf{x})$ out of the $K(\mathbf{x})$ records have the code c_i $(i = 1, \ldots, L)$. As in the duplicate method, we estimate the probability for code c_i in the *NN* approach by $\hat{p}_{nn}(c_i|\mathbf{x}) = k_i(\mathbf{x})/K(\mathbf{x})$. We define the score for the text answer as

$$\gamma(c_i|\mathbf{x}) = \hat{p}_{nn}(c_i|\mathbf{x}) s(\mathbf{x}) \left(\frac{K(\mathbf{x})}{K(\mathbf{x}) + 0.1} \right). \tag{4}$$

The predicted code depends only on $\hat{p}_{nn}(c_i|\mathbf{x})$ because $K(\mathbf{x})$ and $s(\mathbf{x})$ are constant for any given answer text. The role of $s(\mathbf{x})$ and $K(\mathbf{x})/(K(\mathbf{x}) + 0.1)$ is to order observations such that easier-to-classify-answers have a higher score.

The multiplier $s(\mathbf{x})$ makes sense: greater similarity of a new text and its nearest neighbor leads to more accurate classifications. The last term in Equation (4) can be motivated as follows: all else being equal, classification based on a larger number of nearest neighbors will likely be more accurate than that based on fewer nearest neighbors. The multiplier $K(\mathbf{x})/(K(\mathbf{x}) + 0.1)$ equals 0.91 when $K(\mathbf{x}) = 1$ and converges to 1 as $K(\mathbf{x})$ increases. Reflecting lesser importance, this multiplier can, at most, reduce the score by about ten percent, whereas both $\hat{p}_{nn}(c_i|\mathbf{x})$ and s can drive the score to zero. Below, we will show that this works empirically. However, we readily admit this is not the only multiplier that achieves this goal, and that the choice of 0.1 is arbitrary. Using a larger constant extends the range of the multiplier component and thus makes the score more sensitive to $K(\mathbf{x})$. (This is not desirable, as the other two multipliers are more important.)

Table 2. Illustration of calculating $\gamma(c_i|\mathbf{x})$. The unigram variables contain 1 if the word is present in the record and 0 otherwise.

	(Nonzero) ngram variables							
Record	heizung	lüftungsbauer	druck	Occ. Code	$\hat{p}_{nn}(c_i\|\mathbf{x})$	$s(\mathbf{x})$	$\frac{K(\mathbf{x})}{K(\mathbf{x})+0.1}$	$\gamma(c_i\|\mathbf{x})$
Training 1	0	0	1					
Training 2	0	0	1	8251	0.75	0.5774	0.9756	0.4225
Training 3	0	0	1					
Training 4	0	1	0	7136	0.25	0.5774	0.9756	0.1408
Test answer	1	1	1	$\hat{c}_i = 8251$				

For example, the text answer of a new record was "Heizungs und Lüftungsbauer, Drucker". The text consisted of three (stemmed) unigram variables: "heizung" (heating), "lüftungsbau" (ventilation construction) and "druck" (printer). No duplicates existed, but four records in the training data contained one of the three words. Table 2 shows that three out of the four training records had the answer "Drucker" ("druck" in the stemmed ngram representation) with code 8251 and the other had "Lüftungsbauer" ("lüftungsbau" in the stemmed ngram representation) with code 7136. Based on Equation (3), the similarity between the test answer and any of the training records in Table 2 was $\frac{1}{\sqrt{3}\sqrt{1}} = 0.5774$. So the multiplier in Equation (4) is $K(\mathbf{x})/(K(\mathbf{x}) + 0.1) = 4/4.1 = 0.9756$. However, $\hat{p}_{nn}(c_i = 8251|\mathbf{x}) = 3/4$ and $\hat{p}_{nn}(c_i = 7136|\mathbf{x}) = 1/4$. The difference of the γ scores of the two codes was dueto the different probability estimates. In this example, the test answer was assigned code 8251 because it had the largest score ($\gamma = 0.4225$).

4. Occupation Coding for the ALLBUS Survey

We first describe the ALLBUS data (Subsection 4.1) and then show the importance of our definition of duplicates (Subsection 4.2). Next, we compare the proposed automatic coding methods using the ALLBUS data (Subsections 4.3 and 4.4). We conclude with a simulation to explore the influence of duplicates and noise variables in Subsection 4.5.

4.1. Problem and Data

The German General Social Survey (ALLBUS) conducts repeated cross-sectional surveys of the adult German population living in private households, with an oversampling of the residents of East Germany. ALLBUS has been conducted every two years since 1980; initially covering West Germany and expanding to former East Germany after German reunification in 1990 (ALLBUS 2015; Koch and Wasmer 2004). The main topics concern attitudes, behavior, and social structure.

The targeted net sample size is usually 3,500. Since 1994, the samples have been drawn in two stages. In the first stage, about 160 communities (primary sampling units) are selected. In the second stage, addresses of individuals are randomly selected from thelists of residents in every community. Every two years, a fresh probability sample is drawn from the German register. ALLBUS surveys are conducted face-to-face.

ALLBUS interviewers asked about occupation multiple times: current occupation of respondent, last occupation of respondent (if not employed), occupation of spouse

(if married), occupation of partner (if not married but with partner), occupation of father, and occupation of mother. In the ALLBUS survey, the interviewer asks the following questions which are recommended by official statistics in Germany (Statistisches Bundesamt 2010): "What work do you do in your main job? Please describe your work precisely. Does this job, this work have a special name?" (Scholz and Wasmer 2009). Interviewers were free to combine the answers, and were not asked to write one answer after another. The occupation questions for partners/spouses/parents are analogous, using the same format. The answers were pooled to form a single data set. Prior to the open-ended questions about all occupations, respondents were also asked: "Please classify your occupational status according to this list." The list contains 32 occupation statuses in twelve categories. We refer to this below as (self-recorded) occupation status.

The ISCO-88 coding of the text answers was done by GESIS in a two-step procedure. First, automatic coding was attempted using the in-house software, *textpack* (Geis and Hoffmeyer-Zlotnik 2000; Züll 2014). Then, such automatically coded answers were verified by a professional coder. All remaining responses were manually coded in a second step according to an extensive coding manual (Geis 2011). The in-house software used a dictionary with about 4,500 predefined combinations of ISCO codes. Because the dictionary mostly contains duplicates from previous surveys, *textpack* implements the duplicate approach, with additional hand-crafted rules (however, the coder may also override some codes in light of occupational status, education, or other information).

For each word or phrase listed in the dictionary, *textpack* searches for exact matches in the data and outputs the associated code. Such rules were applied one at a time (and the rule order may affect the result). If a rule was matched exactly, a response was coded. If none of the rules applied, it was manually coded by professional coders. Typically, *textpack* coded about 50% of the responses. GESIS used self-reported occupation status only if text was unclear or ambiguous. In the 2006 survey, 9,137 observations were coded into 399 distinct unit occupation codes and 140 minor group codes (see appendix A).

To apply the proposed methods, we encoded text answers into unigram variables (Schonlau and Guenther 2016). All such variables were indicator variables specifying the presence or absence of the corresponding word. We applied stemming, using a German Porter stemmer (Snowball 2015) and removed German "stopwords" as well as punctuation marks. The removal of stopwords and the use of stemming reduced the number of ngram variables. As is standard practice, we also created a variable that counted the number of words contained in the answer. All in all, 4,232 indicator variables were created in addition to the number-of-words variable. In addition to the text response, the survey also contains self-reported occupation status, which was also included among the independent variables.

For a statistical learning approach, we use support vector machines (*SVM*) (Vapnik 2000) with a linear kernel, which has been shown to work well in text categorization (Joachims 1998). The linear kernel requires only a single tuning parameter, C, that controls the trade-off between the training error and model complexity. In this data set, the choice of C had little influence on prediction accuracy and we used $C = 1$ throughout the study. As is common, the *SVM* scores were converted into probabilities using Platt's method (Platt 1999), which performs a regularized logistic regression of class membership on the *SVM* score.

We evaluate the approaches using ten-fold cross validation (*CV*). This means we randomly divide the data into ten equal-sized parts. We use the first nine parts to train the

model, and the last part to test the model. Accuracy is only evaluated on the test data. In turn, we use each of the ten parts as test data and average the results. As a consequence, the size of the training data is therefore 90% of the data, or 8,223 observations. For the purpose of evaluating prediction accuracy we assume that the original codes assigned by GESIS and the professional coders are correct.

The analysis was carried out in *R* (R Core Team 2014), and package *e*1071 (Meyer et al. 2014) is used for the construction of the *SVM* models.

Most open-ended answers were short; 66.5% of the answers consisted of a single word. The median length was one word; the average length was 1.8 words and the maximum length was 17 words. About 60% of the data consisted of (ngram-based) duplicate observations. Among duplicate observations, the median number of duplicates was three, with a higher average (6.8) due to some very frequent duplicates (maximum = 221 duplicates). The text with the most duplicates was "Landwirt" (farmer).

4.2. Ngram Vs. String-Based Definition of Duplicates

The purpose of this section is to demonstrate that the ngram-based method of duplicate is preferable to the string-based methods. Here we explore how much the definition of duplicate mattered for the two best performing methods, NN-3 and hybrid-3/4digit, which are explained later. We compared the ngram-based method with original string (without any processing) and preprocessed string methods. Preprocessed strings refer to lower casing and stripping off leading and trailing spaces in the original strings. As described in Subsection 4.1, ngram variables were obtained after stemming, and removing stopwords and punctuation marks.

The percentage of duplicates is 52.6% for the identical-string-duplicates, 56.7% for the preprocessed-string-duplicates, and 60.0% for the ngram-duplicates. However, the quality of the duplicates did not degrade: identical-string-duplicates (preprocessed-string-duplicates, ngram-duplicates) had identical occupation codes 91.9% (91.6%, 92.0%) of the time. The remaining eight percent represent coders' attempt to recode otherwise unambiguous text in light of occupational status or education. For example, a pharmacist with lower occupational status might be reclassified as pharmaceutical assistant. Of course, misclassification errors are also possible.

Figure 1 shows the trade-off between accuracy and production rate for the three definitions of duplicates for hybrid-3/4digit (left panel) and NN-3 (right panel). The use of the ngram definition of duplicates improved accuracy in both methods for moderate and high production rates. With full automation, accuracy increased from 0.54 (without preprocessed) to 0.65 for the hybrid-3/4digit method, and from 0.47 (without preprocessed) to 0.65 for the NN-3 method. Preprocessed-string-duplicates fare somewhat better than unprocessed strings, but the success of the ngram-based definition clearly goes far beyond string preprocessing.

4.3. Accuracy of the Nearest Neighbor Method

We first investigated the coding performance of the modified *NN* method. The score in Equation (4) has three components. To demonstrate that all three components are helpful, we evaluate both the proposed overall score (NN-3) as well as a reduced score missing one

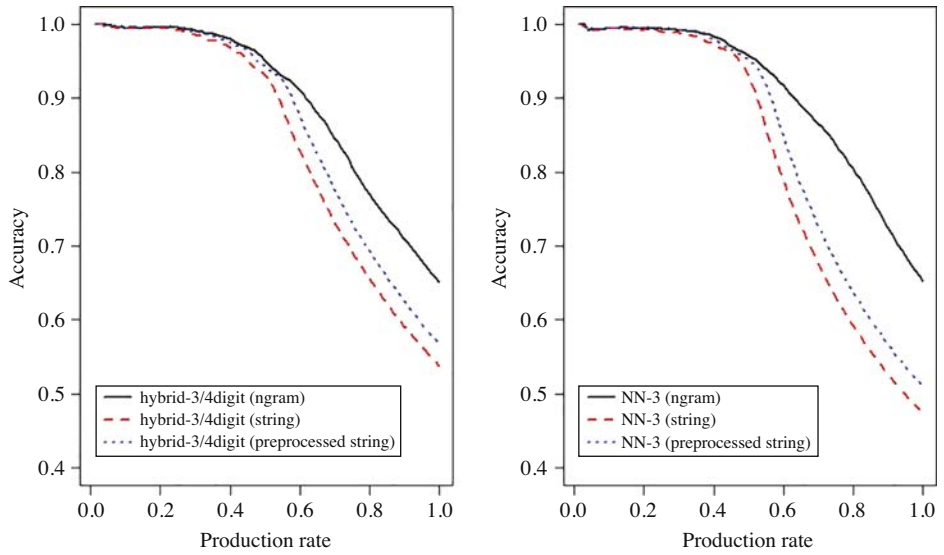

Fig. 1. Accuracy for a given production rate for two approaches based on three different definitions of duplicates "ngram", "string" and "preprocessed string". The left panel shows the results of hybrid-3/4digit and the right panel shows those of NN-3. The "ngram" definition of duplicates is far superior.

(NN-2) or two components (NN-1) with corresponding scores γ_1, γ_2 and γ_3:

$$\begin{aligned}
\text{(NN-1)} \quad \gamma_1 &= \max_i \hat{p}_{nn}(c_i|\mathbf{x}) \\
\text{(NN-2)} \quad \gamma_2 &= \max_i \hat{p}_{nn}(c_i|\mathbf{x})\, s(\mathbf{x}) \\
\text{(NN-3)} \quad \gamma_3 &= \max_i \hat{p}_{nn}(c_i|\mathbf{x})\, s(\mathbf{x}) \left(\frac{K(\mathbf{x})}{K(\mathbf{x}) + 0.1} \right)
\end{aligned}$$

Figure 2 shows the accuracies of each approach as a function of the production rate. (These were average accuracies from the ten-fold cross validation mentioned earlier). Answer texts with higher scores were coded first; a production rate of, say, ten percent refers to coding ten percent of the answer texts with the highest scores automatically. When the production rate equals 100%, the accuracy is the same for all the approaches because the second and third terms in Equation (4) do not affect which code is assigned, but rather are used to prioritize more similar observations and observations with multiple nearest neighbors by assigning them a higher score. Prioritizing affects the accuracy at production rates of less than 100% (because observations with the highest score are chosen first). The improvement from NN-1 to NN-2 showed that similarity s was helpful for finding easier-to-classify-answers. Likewise, the accuracy differences between NN-2 and NN-3 showed that the term $\frac{K(\mathbf{x})}{K(\mathbf{x})+0.1}$ improved the performance at low to medium production rates.

Having established that NN-3 is preferable to NN-1 and NN-2, we next compare NN-3 with all other approaches.

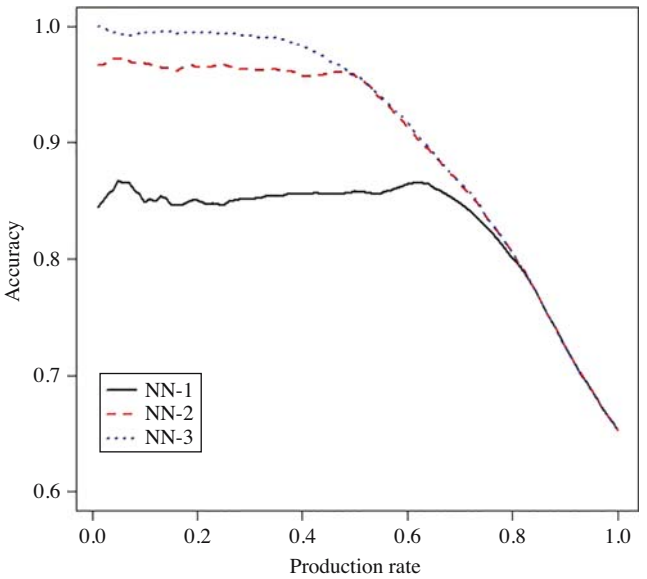

Fig. 2. Accuracy of three variations on the nearest neighbor approach as a function of production rates. NN-1, NN-2, and NN-3 refer to scores using $\gamma_1 = \hat{p}_{nn}(c_i|\mathbf{x})$, $\gamma_2 = \hat{p}_{nn}(c_i|\mathbf{x})s$ *and* $\gamma_3 = \hat{p}_{nn}(c_i|\mathbf{x})s\left(\frac{K(\mathbf{x})}{K(\mathbf{x})+0.1}\right)$, *respectively.*

4.4. Comparison of Methods

Here we compare the accuracy as a function of production rate for the proposed methods (hybrid-4digit, hybrid-3/4digit, and NN-3) as well as some default methods (duplicate method, svm-4digit, svm-3/4digit). The duplicate method refers to assigning the code of ngram duplicates (or a random code if no duplicates exist), svm-4digit refers to an *SVM* model based on four-digit occupation codes. The svm-3/4 digit refers to an *SVM* model based on averaged probability from separate models for three-digit and four-digit occupation codes as described in Equation (1). For all methods, a production rate of x% refers to the x% of the data that have the highest score (or probability).

Figure 3 shows the accuracy as a function of the production rate for the different methods. For all methods, there were trade-offs between the accuracy and the production rate. The modified nearest neighbor method, NN-3, performs equal to or slightly better than the next best method, hybrid-3/4digit. NN-3, hybrid-4digit, and hybrid-3/4digit uniformly beat the duplicate method and both svm methods.

A production rate of 100% corresponds to classifying all answers automatically. At full automation, NN-3 and hybrid-3/4digit perform equally well. At full automation, svm-3/4digit has an accuracy of 59%, the duplicate method has an accuracy of 53%, and the hybrid-3/4digit method increases the accuracy to 65%.

Figure 3 also shows the duplicate accuracy remained at around 95% up to a production rate of about 0.55. About 55% of the test data in any given cross-validation were duplicates and thus duplicates were used for coding. However, when no duplicates exist in the training data, the duplicate approach assigned equal probabilities to all codes, resulting in the random code assignment and accuracy near zero. The accuracy started decreasing at a production rate of around 0.55, from which no additional records of some *CV* test samples

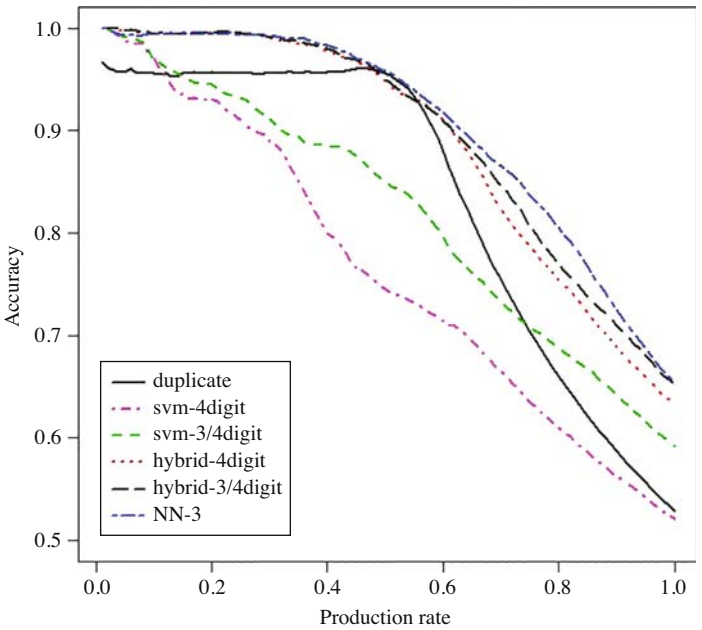

Fig. 3. Comparison of different methods for occupation coding. Methods include statistical learning (svm-4digit), statistical learning from two models at different levels of aggregation (svm-3/4digit), and two hybrid methods combining duplicate-predictions with svm-4digit and svm-3/4digit, respectively.

could be classified by the method. From a production rate of 0.60, all of the *CV* test data sets had no duplicates and the method performed poorly. NN-3, hybrid-4digit, and hybrid-3/4digit beat the duplicate method even for production ranges where duplicates are available.

Combining the four-digit unit and three-digit minor code methods (svm-3/4digit) was uniformly superior to using the unit code method only (svm-4digit). For example, for fully automated coding, the accuracy for svm-3/4digit was 0.59, as compared with 0.52 for svm-4digit. The hybrid approaches performed very similarly up to a production rate of about 60%. After that, the hybrid-3/4digit performs a little better than hybrid-4digit. When duplicates were available for hybrid-3/4digit, the predicted codes mostly agreed (83%) with those predicted by the duplicate method.

The performances of hybrid-3/4digit and the NN-3 were similar for fully-automated coding as well as at low-medium production rates. NN-3 appeared to slightly outperform hybrid-3/4digit at medium-high production rates.

The curves in Figure 3 help us decide which texts should be classified automatically and which should be classified manually. For example, if the client decides that 80% accuracy is required, then Figure 3 suggests that 76% of the data can be classified automatically with the hybrid method and 81% with the NN-3 method. Relative to applying the duplicate-based approach, this increases production from about 58% to 76% or 81%.

4.5. Simulation

The purpose of this section is to explore to what extent the methods are robust to possible idiosyncrasies of the data. We considered two possible concerns with our example data:

1) The data contain a large percentage (50%) of duplicates. 2) The text answers are unusually clean and contain fewer superfluous words than usual.

In the first case, in the context of occupation coding a large number of duplicates is very common. (Duplicates here refers to ngram duplicates). To simulate a data set with fewer duplicates, a random subset of duplicate records was removed so that in the reduced data only about ten percent duplicates of the test records had duplicates. The reduced data set contained 4,722 observations.

As expected, Figure 4 shows that the accuracy (for a given production rate) for all methods decreased for this much more difficult problem. The relative performance of the methods is very similar with one notable exception: previously, both NN-3 and hybrid3/4-digit performed similarly. Now, NN-3 clearly outperforms the hybrid-3/4digit method. The NN-3 method remains superior to NN-1 and NN-2 analogous to Figure 2 (The analogous figure is not shown).

In the second case, less clean text answers would have resulted in additional words that are not related to the occupation code. Such additional words translate into indicator variables (presence or absence of the word) in the data. There are typically many such variables, each with a low probability. We added 100 independent "noise" indicator variables to the data. Each variable followed a Bernoulli distribution with an 0.01 probability of success.

The results are shown in Figure 5. Adding the noise variables decreased the number of duplicates. Hence the accuracy of the duplicate method started decreasing at a production rate of around 0.2 instead of around 0.55. The results lead to roughly the same conclusions as we obtained from Figures 3 and 4. NN-3 and hybrid-3/4digit were comparable, with NN-3 having a slight edge at lower production rates.

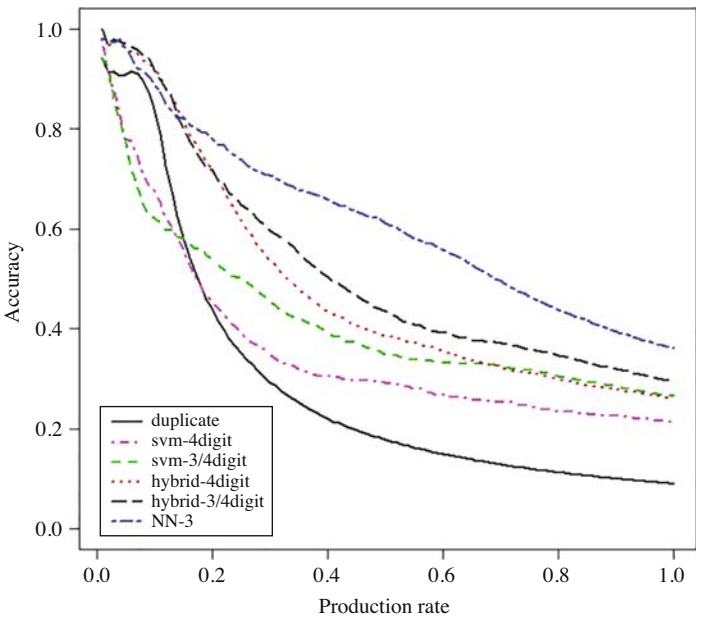

Fig. 4. Comparison of the same methods as in Figure 3 on a reduced data set containing only ten percent duplicates.

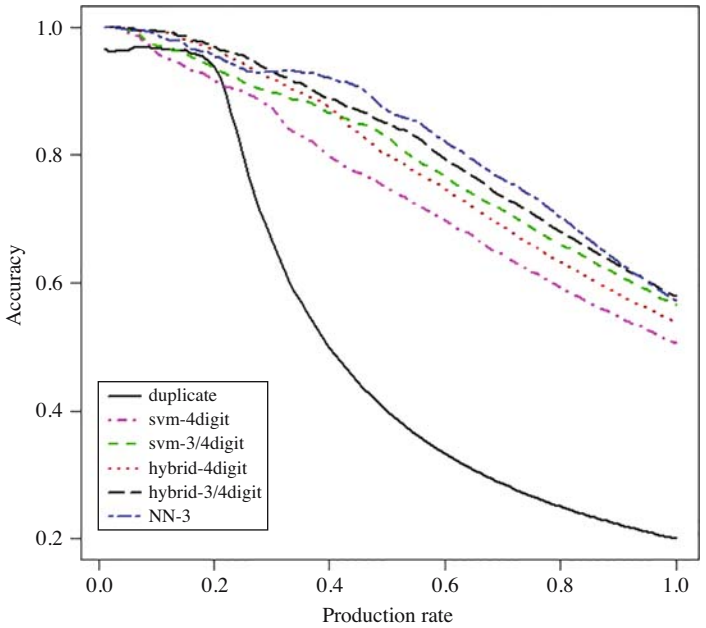

Fig. 5. Comparison of the same methods as in Figure 3 with 100 noise variables added to the data.

5. Discussion

We have investigated several novel approaches for automated occupation coding for any desired production rate. The two best-performing methods, the modified nearest neighbor method (NN-3) and a hybrid method (hybrid-3/4digit) substantially improve the accuracy compared with both statistical learning (*SVM* in the example) by itself and the duplicate method at any production rate in the ALLBUS data. As the percentage of duplicates decreases, a simulation shows that NN-3 gains a relative advantage over the hybrid method.

Either accuracy or production rate can be set at a target rate which determines the second measure. For example, targeting 80% accuracy for the automated coding, the hybrid-3/4digit and NN-3 approaches could categorize 76% and 81% of the data automatically, while the numbers obtained by the *SVM* and duplicate methods individually were 60% and 66%, respectively. If production rate is fixed at 80%, the hybrid-3/4digit and NN-3 could achieve an accuracy of 77% and 81%, while the *SVM* and duplicate approaches reported accuracy of 69% and 66%. Note that accuracy for each category may differ from the overall accuracy. Categories that contain more hard-to-code answers than others achieve lower accuracies.

In addition, we have learned:

(1) Even at low production rates when duplicates exist, NN-3 and hybrid achieve a higher accuracy than the duplicate method.
(2) Using the duplicate method where duplicates exist and using statistical learning otherwise is not the best strategy (Figure 3 shows the proposed methods beat the duplicate method where duplicates exist.). We instead recommend the hybrid method that integrates the two approaches.

(3) Combining aggregate and detailed learners improves accuracy for some learning algorithms. For example, where svm-4digit and svm-3/4digit disagree in the ALLBUS data, svm-3/4digit is correct 87% of the time.

Why do the NN-3 and hybrid methods beat *SVM* and the duplicate approach? Because a duplicate is also a nearest neighbor, both methods rely on nearest neighbors. Nearest neighbor algorithms are effective when prediction is highly local and little can be gained from observations further away. This may explain why NN-3 and hybrid methods beat *SVM*, one of best statistical learning algorithms in existence. Both proposed methods beat the duplicate approach because a) they both can distinguish between easier-to-code and harder-to-code duplicates leading to higher accuracies at lower production rates, b) the hybrid- 3/4 method can break ties among duplicates, and c) the duplicate approach performs poorly when no duplicates exist.

The NN-3 approach can be computationally expensive when the training data set is very large. The hybrid method requires finding duplicates, but on the other hand, finding duplicates is much less expensive because it does not require a sorting step.

We have combined the aggregate method with the hybrid method, leading to better results. The modified nearest neighbor method could also be combined with the idea of aggregating different level scores. However, the resulting method showed almost the same performance as NN-3.

We now comment on the importance of some data analysis choices. First, duplicates were defined as having the same ngram representation rather than being identical strings. This increased the number of duplicates and substantially improved accuracy at moderate and high production levels. Second, self-reported occupation status (STIB) was used as a covariate for statistical learning. We found that including STIB made little difference. Third, we supported German language stemming, but it turned out this had almost no effect. Because the text was written by interviewers (rather than respondents) our data were relatively clean with many one-word answers. Stemming is likely more important with messier data.

We next comment on possible limitations arising from idiosyncrasies of the ALLBUS data set. The proposed methods are not limited to the ISCO-88 coding scheme. One of the methods relies on a hierarchical coding scheme, but all occupation codes are hierarchical. We have analysed 9,137 observations. While this data set is probably larger than most data sets analysed in statistical journals, at national statistics agencies far larger data sets arise sometimes with millions of observations. The proposed methodology is not limited to a specific data size, but it is unclear whether the performance of the proposed methodology relative to the alternative algorithms would be equally impressive with millions of observations. We have pooled self-recorded occupations and occupations from partners, spouses, and parents. We investigated whether this distorted results somehow. Specifically, we reduced the data set to one occupation question per respondent. We found this did not meaningfully affect the results.

For the hybrid method, we used *SVM* as the statistical learning method of choice. While *SVM* is one of best performing methods available, other statistical learning methods could be chosen, provided that they output a probability (or a score that can be transformed into a pseudo-probability) rather than just a classification. Naturally, better predictions from the

statistical learning method will tend to improve the hybrid method also, particularly when there are no duplicates.

All proposed approaches rely on training data. For statistical learning, the size of the training data needs to be large relative to the number of occupation codes. In the ALLBUS data, the size of the training data (implied by cross-validation) was 8,226. Relative to the 399 occupation codes, this is an average of 20.6 observations per code. More training data will tend to increase the number of duplicates.

Cross-validation deals with unseen data, but does not take into account time trends. To the extent that language use changes from year to year, any classifier would slowly degrade over time.

In summary, we proposed new approaches to automated occupation coding that lead to vastly improved coding accuracy at both high and low production rates in our example data. While not conclusive, this bodes well for other occupation data sets.

Appendix A

There are more distinct codes in the GESIS data than the 390 ISCO-88 unit codes for several reasons: 1) When there is sufficient information to identify a minor group, but not sufficient information to identify a unit code, the minor code is used and a zero is appended (e.g., minor group 112 would turn into 1120). 2) Sometimes a minor group can be identified and the text is specific enough to identify the exact occupation, but that occupation is not listed. In that case a separate code is used ending in a nine (e.g., 1129 in the previous example) 3) ISCO-88 allows users to define additional codes for occupations that are not explicitly mentioned. GESIS has defined 10 such codes (e.g., housewife, not codable, don't know). The total of possible GESIS codes is 641 (390 unit codes $\pm$ 116 minor groups $\pm$ 28 sub-major groups $\pm$ 10 major groups $\pm$ 10 GESIS specific codes $\pm$ 87 codes for occupations not elsewhere classified). In the ALLBUS 2006 survey 399 of the 641 distinct codes were observed.

6. References

ALLBUS. 2015. Available at: http://www.gesis.org/allbus (accessed October 10, 2016).

Appel, M.V. and E. Hellerman. 1983. "Census Bureau Experiments with Automated Industry and Occupation Coding." In Proceedings of the American Statistical Association, Section on Survey Research Methods. August 15–18, 1983, Toronto, Canada. 32–40.

Belloni, M., A. Brugiavini, E. Meschi, and K. Tijdens. 2014. *Measurement Error in Occupational Coding: an Analysis on SHARE Data*. Ca' Foscari University of Venice, Department of Economics, Working Paper 24. Doi: http://dx.doi.org/10.2139/ssrn.2539080.

Bethmann, A., M. Schierholz, K. Wenzig, and M. Zielonka. 2014. "Automatic Coding of Occupations." In Proceedings of Statistics Canada Symposium. August 29–31, 2014, Québec, Canada. Available at: http://www.statcan.gc.ca/sites/default/files/media/14291-eng.pdf (accessed October 10, 2016).

Chen, B.-C., R.H. Creecy, and M.V. Appel. 1993. "Error Control of Automated Industry and Occupation Coding." *Journal of Official Statistics* 9: 729–745. http://www.jos.nu/Articles/abstract.asp?article=94729 (accessed October 10, 2016).

Clarke, F.R. and S.J. Brooker. 2011. Use of Machine Learning for Automated Survey Coding. In Proceedings of the 58th ISI World Statistics Congress. August 21–26, 2011, Dublin, Ireland.

Conrad, F.G., M.P. Couper, and J.W. Sakshaug. 2016. "Classifying Open-Ended Reports: Factors Affecting the Reliability of Occupation Codes." *Journal of Official Statistics* 32: 75–92. Doi: http://dx.doi.org/10.1515/JOS-2016-0003.

Creecy, R.H., B.M. Masand, S.J. Smith, and D.L. Waltz. 1992. "Trading MIPS and Memory for Knowledge Engineering." *Communications of the ACM* 35: 48–64. Doi: http://dx.doi.org/10.1145/135226.135228.

Day, J. 2014. *Using an Autocoder to Code Industry and Occupation in the American Community Survey*. Presentation for the Federal Economic Statistics Advisory Committee Meeting. Available at: http://www2.census.gov/adrm/fesac/2014-06-13_day.pdf (accessed October 10, 2016).

Elias, P. 1997. "Occupational Classification (ISCO-88): Concepts, Methods, Reliability, Validity and Cross-National Comparability." OECD Labour Market and Social Policy Occasional Papers 20, OECD Publishing. Available at: https://ideas.repec.org/p/oec/elsaaa/20-en.html (accessed October 10, 2016).

Elias, P. and M. Birch. 2010. *Tuning CASCOT for Industry and Occupation Coding in the Scottish Census of Population 2011*. Technical Report, Institute for Employment Research. Coventry: University of Warwick.

Ferrillo, A., S. Macchia, and P. Vicari. 2008. "Different Quality Tests on the Automatic Coding Procedure for the Economic Activities Descriptions." In Proceedings of the European Conference on Quality in Official Statistics – Q2008. July 8–11, 2008, Rome, Italy. Available at: http://q2008.istat.it/sessions/paper/15Ferrillo.pdf (accessed January 2017).

Fix, E. and J.L. Hodges. 1951. *Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties*. Technical Report, USAF School of Aviation Medivine, Randolph Field, Texas. Project 21-49-004, Rept. 4, Contract AF41(128)-31, February 1951.

Friedman, J.H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." *The Annals of Statistics* 29: 1189–1232. Available at: http://www.jstor.org/stable/2699986 (accessed October 10, 2016).

Ganzeboom, Harry B.G. and Donald J. Treiman. 2003. "Three Internationally Standardised Measures for Comparative Research on Occupational Status." In *Advances in Cross-National Comparison: A European Working Book for Demographic and Socio-Economic Variables*, edited by J.H.P. Hoffmeyer-Zlotnik and C. Wolf, pp. 159–193. Doi: http://dx.doi.org/10.1007/978-1-4419-9186-7_9.

Geis, A. 2011. *Handbuch für die Berufsvercodung*. Technical Report, GESIS, Mannheim, Germany. Available at: http://www.gesis.org/fileadmin/upload/dienstleistung/tools_standards/handbuch_der_berufscodierung_110304.pdf (accessed October 10, 2016).

Geis, A.J. and J.H.P. Hoffmeyer-Zlotnik. 2000. "Stand der Berufsvercodung." *ZUMA Nachrichten* 24: 103–128.

Iezzi, D.F., M. Lori, F. Lorenzini, M. Nicosia, and S. Stoppiello. 2014. "An Application of Text Mining Technique for the Census of Nonprofit Institutions." In *Statistical Methods and Applications from a Historical Perspective*, edited by F. Crescenzi and S. Mignani, pp. 143–152. Springer. Doi: http://dx.doi.org/10.1007/978-3-319-05552-7_13.

International Labour Office. 1990. International Standard Classification of Occupations, ISCO-88. International Labour Office. Available at: http://www.ilo.org/public/libdoc/ilo/1990/90B09_411_engl.pdf (accessed October 10, 2016).

Joachims, T. 1998. "Text Categorization with Support Vector Machines: Learning with Many Relevant Features." In Proceedings of the 10th European Conference on Machine Learning, Volume 1398. April 21–23, 1998, Chemnitz, Germany, 137–142. Doi: http://dx.doi.org/10.1007/BFb0026683.

Jones, R. and P. Elias. 2004. *CASCOT: Computer-Assisted Structured Coding Tool*. Technical Report, Institute for Employment Research. Coventry: University of Warwick. Available at: http://www2.warwick.ac.uk/fac/soc/ier/publications/software/cascot/ (accessed October 10, 2016).

Jung, Y., J. Yoo, S.-H. Myaeng, and D.-C. Han. 2008. "A Web-Based Automated System for Industry and Occupation Coding." In *Web Information Systems Engineering - WISE 2008*, edited by J. Bailey, D. Maier, K.-D. Schewe, B. Thalheim, and X. Wang. Volume 5175, 443–457. Springer. Doi: http://dx.doi.org/10.1007/978-3-540-85481-4_33.

Kalpic, D. 1994. "Automated Coding of Census Data." *Journal of Official Statistics* 10: 449–463.

Knaus, R. 1987. "Methods and Problems in Coding Natural Language Survey Data." *Journal of Official Statistics* 3: 45–67.

Koch, A. and M. Wasmer. 2004. "Der ALLBUS als Instrument zur Untersuchung sozialen Wandels: Eine Zwischenbilanz nach 20 Jahren." In *Sozialer und Politischer Wandel in Deutschland*, edited by R. Schmitt-Beck, M. Wasmer, and A. Koch, 13–41. VS Verlag für Sozialwissenschaften.

Maitra, R. and I.P. Ramler. 2010. "A k-mean-directions Algorithm for Fast Clustering of Data on the Sphere." *Journal of Computational and Graphical Statistics* 19: 377–396. Doi: http://dx.doi.org/10.1198/jcgs.2009.08155.

Meyer, D., E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch. 2014. *e1071: Misc Functions of the Department of Statistics, TU Wien*. Available at: http://CRAN.R-project.org/package=e1071 (accessed October 10, 2016).

O'Reagan, R.T. 1972. "Computer-Assigned Codes from Verbal Responses." *Communications of the ACM* 15: 455–459. Doi: http://dx.doi.org/10.1145/361405.361419.

Ossiander, E.M. and S. Milham. 2006. "A Computer System for Coding Occupation." *American Journal of Industrial Medicine* 49: 854–857. Doi: http://dx.doi.org/10.1002/ajim.20355.

Platt, J. 1999. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods." In *Advances in Large Margin Classifiers*, edited by A.J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, 61–74. Cambridge, Massachusetts: MIT Press.

R Core Team. 2014. "R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing." Available at: http://www.R-project.org/ (accessed October 10, 2016).

Russ, D.E., K.-Y. Ho, C.A. Johnson, and M.C. Friesen. 2014. "Computer-Based Coding of Occupation Codes for Epidemiological Analyses." In Proceedings of the 27th IEEE International Symposium on Computer-Based Medical Systems. May 27–29, 2014, New York, USA, 347–350. Doi: http://dx.doi.org/10.1109/CBMS.2014.79.

Schierholz, M. 2014. "Automating Survey Coding for Occupation." Master's thesis, Ludwig-Maximilians-Universität Munich. Available at: https://epub.ub.uni-muenchen.de/21444/index.html (accessed October 10, 2016).

Scholtus, S., R. van de Laar, and L. Willenborg. 2014. *The Memobust Handbook on Methodology for Modern Business Statistics*. Available at: https://ec.europa.eu/eurostat/cros/system/files/NTTS2013fullPaper_246.pdf (accessed January 2017).

Scholz, E., and M. Wasmer. 2009. *German General Social Survey 2006. English Translation of the German "ALLBUS"- Questionnaire*. Technical Report, GESIS, Mannheim, Germany. Available at: http://nbn-resolving.de/urn:nbn:de:0168-ssoar-207035 (accessed October 10, 2016).

Schonlau, M., and N. Guenther. 2016. Text Mining Using N-Grams. *Social Science Research Network*. Doi: http://dx.doi.org/10.2139/ssrn.2759033.

Silla, C.N., and A.A. Freitas. 2011. "A Survey of Hierarchical Classification across Different Application Domains." *Data Mining and Knowledge Discovery* 22: 31–72. Doi: http://dx.doi.org/10.1007/s10618-010-0175-9.

Snowball. 2015. Available at: http://snowball.tartarus.org/algorithms/german/stemmer.html (accessed October 10, 2016).

Statistisches Bundesamt. 2010. *Demographische Standards*. Technical Report, Wiesbaden, Germany. Available at: https://www.destatis.de/DE/Methoden/StatistikWissenschaft-Band17.html (accessed October 10, 2016).

Thompson, M., M.E. Kornbau, and J. Vesely. 2012. "Creating an Automated Industry and Occupation Coding Process for the American Community Survey." Available at: http://ftp.census.gov/adrm/fesac/2014-06-13_thompson_kornbau_vesely.pdf (accessed October 10, 2016).

Tijdens, K. 2014. "Dropout Rates and Response Times of an Occupation Search Tree in a Web Survey." *Journal of Official Statistics* 30: 23–43. Doi: http://dx.doi.org/10.2478/jos-2014-0002.

Tijdens, K. 2015. "Self-Identification of Occupation in Web Surveys: Requirements for Search Trees and Look-Up Tables." *Survey Methods: Insights from the Field (SMIF)*. Doi: http://dx.doi.org/10.13094/SMIF-2015-00008.

Tourigny, J.Y., and J. Moloney. 1995. "The 1991 Canadian Census of Population Experience with Automated Coding." In United Nations Statistical Commission on Statistical Data Editing.

Vapnik, V.N. 2000. *The Nature of Statistical Learning Theory*. 2nd edition. New York: Springer.

Weiss, S.M., N. Indurkhya, T. Zhang, and F. Damerau. 2010. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. New York: Springer.

Wenzowski, M.J. 1988. "ACTR – A Generalised Automated Coding System." *Survey Methodology* 14: 299–308.

Yu, C. 2002. *High-Dimensional Indexing: Transformational Approaches to High-Dimensional Range and Similarity Searches*. Volume 2341. Berlin: Springer. Doi: http://dx.doi.org/10.1007/3-540-45770-4.

Züll, C. 2014. *Berufscodierung*. Technical Report, GESIS – Leibniz Institut für Sozialwissenschaften (SDM Survey Guidelines). Mannheim. Doi: http://dx.doi.org/10.15465/sdm-sg_019.

Received March 2016
Revised October 2016
Accepted October 2016